REMARKS

The present amendment is submitted in response to the Office Action mailed January 11, 2008. Claims 1-6 are pending in this application. In view of the remarks to follow, reconsideration and allowance of this application are respectfully requested. Applicant appreciates the courtesy granted to Applicant's attorneys, George Kaplan (Reg. No. 28,375) and Michael A. Scaturro (Reg. No. 51,356), during a telephonic interview conducted on Monday, March 31, 2008. During the interview, the merits of Applicants' proposed new limitation to claim 1 were discussed. Applicants' representatives elaborated on differences between the current claim limitations and cited passages of the Wagner and Ko references. A general agreement was reached to file an amendment which may include more limitations beyond those presently proposed. The Examiner agreed to reconsider the current rejection in view of Applicants' arguments.

The Invention

Prior to discussing substantive issues related to the instant amendment, it is instructive to briefly review the invention. As discussed at page 9 of the specification, conventional static heuristic analysis determines whether codes for performing self-replication exist based <u>only</u> on the presence of a method call sequence usable for the self-replication. The method of the present invention goes <u>beyond</u> this simple analysis by referencing the parameters and return values of the method sequence constructing the malicious behavior. More particularly, the method of the invention models malicious behavior as a <u>combination</u> of unit behaviors, which are <u>further</u> comprised of sub-unit behaviors and method calls. The unit behaviors are converted to matching rules and relation rules for defining sentence types to be identified in a suspected script code (i.e., the matching rule) and for defining a relation between matched patterns (i.e., the relation rule). Using the matching rule and relation rule, instances of the matching rule are generated from the script code that match the matching rule. The instances of the matching rule are then analyzed to generate instances of the relation rule by searching for instances satisfying the relation rule from a set of the generated instances of the matching rule.

35 U.S.C. §103(a)

In the Office Action, Claims 1-3 were rejected under 35 U.S.C. §103(a) as being allegedly unpatentable over "Intrusion Detection Using Static Analysis" (hereinafter "Wagner") in view of "Static Analysis" (hereinafter "Webb") and further in view of U.S Patent No. 6,697,950 – issued to Ko.

In making the rejection, the Examiner contends regarding claim 1, Wagner discloses a method for detecting malicious scripts using a static analysis, comprising the step of: classifying, by modeling a malicious behavior in such a manner that it includes a combination of unit behaviors each of which is composed of sub-unit behaviors or one or more method calls, each unit behavior and method call sentence into a matching rule for defining a relation between patterns matched so that the malicious behavior can be searched by rule variables used in the sentences satisfying the matching rule.

In support of his position, the Examiner directs Applicants' attention to section 4.3, "The abstract stack model", and particularly pages 160-161, "The context-free model") page 158, first paragraph of Wagner and further asserts Wagner discloses generating instances of the relation rule by searching for instances satisfying the relation rule from a set of the generated instances of the matching rule. However, the Examiner admits Wagner does <u>not</u> disclose extracting parameters of functions used in the searched code patterns, and storing the extracted parameters in the rule variables, preferring instead to implement a simpler model.

Wagner teaches a sequence of models proposed to specify expected application behavior, namely, a trivial model, a callgraph model, an abstract stack model, a context-free model and a low-overhead digraph model and the abstract stack model is a refinement of the callgraph model which is itself a refinement of the trivial model. In the callgraph model, the *ordering* of all possible system calls is taken into account in the modeling process. For ease of model generation, Wagner uses an equivalent representation of the model as a non-deterministic finite automation (NDFA). Wagner further discloses the model is a simple application of control-flow analysis. In this regard, Wagner builds a control flow graph associated with the program source code. Each node of the graph is assumed to execute at most one system call. The abstract model merely refines this process by excluding impossible paths. While Wagner checks whether a series of methods constructing a malicious code pattern exist, via the control flow graph and analysis, Wagner does not determine whether parameters and return values associated between the methods match each other, as recited in Claim 1.

Wagner merely teaches whether a series of methods constructing a malicious code pattern exists and does not teach the steps of converting unit behaviors and method call sentences into a matching rule for defining sentence types to be detected in script codes and a relation rule for defining a relation between rule variables used in the sentences satisfying the matching rule, as recited in Claim 1. It therefore follows Wagner does not teach a matching rule comprised of rule-identifiers and patterns to be detected, as recited in Claim 1.

In the Office Action, Webb is cited for allegedly curing the deficiency in Wagner. More particularly, Webb is cited for disclosing the step of extracting parameters of functions used in the searched code patterns, and storing the extracted parameters in the rule variables. Applicants respectfully disagree. There is no teaching in Webb of storing extracted parameters in the rule variables. Webb only teaches, in broad terms, the ability to statically analyze "local variables, data structures, and all other data flow" in a script to determine if the script is non-hazardous. Webb provides no particularity how to statically analyze "local variables, data structures, and all other data flow" to extract parameters of functions used in the searched code patterns.

The Examiner admits neither Wagner or Webb explicitly disclose checking whether parameters and return values associated with the methods match each other and cites Ko to allegedly cure this deficiency. In particular, the Examiner cites Ko for allegedly teaching an analogous static analyzer employing this limitation and refers to Col. 6, lines 1-50 for support. Applicants respectfully disagree and submit Ko does not cure the deficiency of Wagner or Webb.

Ko, at step 410 of the flowchart of FIG. 4, teaches an analyzer 212 checks for suspect macro operations, by comparing the macro operations (including the associated values for variables) against a profile containing information about suspect macro operations and associated values for variables. However, Claim 1 recites a step of checking whether a series of methods constructing a malicious code pattern exist and whether parameters and return values associated between the methods match each other (not a profile). It is respectfully submitted comparing macro operations against a profile, as taught in Ko, is different from checking whether parameters and return values associated between the methods match each other, as recited in Claim 1.

The Examiner also referred to Col. 5 of Ko during the telephone interview, alleging Ko also teaches profile database may also include rules. These rule descriptions have no relevance to the instant application. For example, the rules may determine whether a document is "safe" (Column 5, lines 42-43). It is respectfully submitted such rules do not teach or disclose a step of checking whether parameters and return values associated between the methods match each other, as recited in Claim 1.

In accordance with the method of the invention, parameters and return values, as used in the script sentences, are preferably replaced by "rule variables" for ease of analyzing relations between the rule variables in those sentences that satisfy the matching rule. None of the cited references, alone or in combination, teach this feature.

While Applicants traverse the rejection, Claim 1 has also been amended to recite limitations and/or features which are not disclosed by the cited references, alone and in any reasonable combination.

Claim 1 now recites –

1. A method for detecting malicious scripts using a static analysis, comprising the step of:

checking a script to determine whether a series of methods constructing a malicious code pattern exist and whether parameters and return values associated between the methods match each other,

wherein the checking step comprises the steps of:

a) classifying, by modeling a malicious behavior to include unit behaviors each of which is composed of sub-unit behaviors or one or more method calls,

b) generating a matching rule by converting each identified unit behavior and method call sentence into said matching rule for defining sentence types to be detected in script codes, said matching rule comprising rule identifiers and sentence patterns to be detected and

c) generating at least one relation rule for defining a relation between rule variables used in the sentences satisfying the matching rule;

d) generating instances of the matching rule by:

i) searching for code patterns matched with the matching rule from a relevant script code to be detected,

ii) extracting parameters of functions used in the searched code patterns; and

iii) storing the extracted parameters in the rule variables; and

e) generating instances of the relation rule by searching for instances of the matching rule satisfying the relation rule from the set of the generated instances of the matching rule.

Accordingly, it is believed Claim 1, as amended, recites patentable subject matter; therefore, withdrawal of the rejections with respect to Claim 1 and allowance thereof is respectfully requested. Claims 2 and 3 depend from Claim 1, and therefore include the limitations of Claim 1. Hence, for the same reasons given above for Claim 1, Claims 2 and 3 are believed to contain patentable subject matter. New Claims 4-6 have been added to further clarify the inventive method over the cited references and find clear support throughout the present application and drawings.

Accordingly, in view of the forgoing amendment, accompanying remarks and telephone interview, it is respectfully submitted all claims pending herein are in condition for allowance. Please contact the undersigned attorney should there be any questions. A petition for an automatic one-month extension of time for response under 37 C.F.R. §1.136(a) is enclosed in triplicate, together with the requisite petition fee.

Early favorable action is earnestly solicited.

Respectfully submitted,

George M. Kaplan
Reg. No. 28,375
Attorney for Applicant(s)

DILWORTH & BARRESE, LLP
333 Earle Ovington Boulevard
Uniondale, New York 11553
Tel. No. (516) 228-8484
Facsimile (516)228-8516